Describing demographic models is hard

Aaron P. Ragsdale[1], Dominic Nelson[1], Simon Gravel[1,⋆], and Jerome Kelleher[2,⋆]

[1]McGill University and Genome Québec Innovation Centre, McGill University, Montréal, Québec, Canada

[2]Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford, Oxford, United Kingdom

⋆Joint senior authors, listed alphabetically

May 20, 2020

APR: title thoughts:
Reducing errors in demographic model design and implementation
Lessons learned from misspecified models of human history
Lessons learned from buggy models of human history
Lessons learned from bugs in population genetics models

Abstract

Simulation plays a central role in population genomics studies. Recent years have seen rapid improvements in software efficiency that make it possible to simulate large genomic regions for many individuals sampled from large numbers of populations. The increase in complexity of possible demographic models also provides additional ways that we can get their implementation wrong. Here we describe two errors made in defining population genetic models using the `msprime` coalescent simulator that have found their way into the published record. We discuss how these errors have affected analyses and suggest recommendations for software developers and users to reduce the risk of such errors.

1 Introduction

The `msprime` coalescent simulator (Kelleher et al., 2016; Nelson et al., 2020; Kelleher and Lohse, 2020) is now widely used. The large increases in efficiency over the classical `ms` program (Hudson, 2002) make it feasible to simulate large samples of whole chromosomes for the first time. Another distinct advantage of `msprime` is the Python API that is its primary interface, greatly increasing the flexibility and ease of use over the standard approach of text-based command line interfaces. In particular, programs like `ms` required users to specify cryptic command line options to describe demographic models JK: maybe include an example ms command line?. Particularly for the large models we use today, these are not intuitive or comprehensible for humans. The Python API for `msprime` is a great improvement, allowing the user to state models in a documented and reproducible manner. JK: maybe show the same model as described above in msprime notation?

Implementing multi-population models of demographic history is still hard and error prone, however. In this note we discuss two implementation errors that arose through unfortunate design decisions in `msprime`'s demography API and which then found their way into the scientific record. While one of these errors is not expected to have produced qualitative mistakes, the second lead to more severe ramifications APR: departures?. In light of these implementation errors, we discuss improvements to `msprime`'s API motivated by these discoveries and best practices for implementing and simulating complex multi-population demography.

2 A bad tutorial example

To illustrate the demography API, msprime included a description of a widely-used three population Out-of-Africa model (Gutenkunst et al., 2009) as part of its tutorial documentation. In this model (Fig. 1A), Eurasian (CEU and CHB) and African (YRI) populations split from each other in the deep past, followed by a more recent split of European and Asian populations, with variable rates of continuous migration between each of the populations. However, the implementation in the msprime tutorial was incorrect. Before the time of the split of African and Eurasian populations, when there should have been just a single randomly mating population, migration was allowed to occur between the ancestral population and a second population with size equal to the Eurasian bottleneck size for all time into the past (Fig. 1B). This incorrect model was introduced into the tutorial for msprime version 0.3.0 and remained in the documentation for around four years.

Fortunately, the effects of this error are subtle and it would take an exceedingly sensitive method to detect the difference. Population sizes and structure since the time of the split are unaffected, so that differences in expected F_{ST} are negligible between the correct and incorrect model. However, the ancient structure distorts the distribution of T_{MRCA} beyond the split date and increases heterozygosity in contemporary populations roughly 4% over expectations from the correct model (Fig. 1E-F).

It is fortunate that the error in the model description has quite a small effect because the tutorial code has been copied many times and used in publications. By searching for some identifying strings from the model definition on GitHub, we found 32 repositories containing either direct copies of the erroneous model code, or code that was obviously derived from it. (We have opened issues on each of these repos to alert the authors.) In most cases the publications use simulations to test some inference method which is not directly concerned with detailed demography, and the model is simply used as an example of a complex population history (Kelleher et al., 2019; Albers and McVean, 2020; Tong and Hernandez, 2020). Zhou et al. (2018) use the incorrect model as an example of their method for visualising demographic models can support msprime input. Finally, Pfaffelhuber et al. (2020) use simulations of the incorrect model demography to evalute their method for choosing ancestry informative markers. Given the very subtle effect of the incorrect model on demography (and the fact the method was evaluated using other simulations and real data), it seems unlikely that the model details will have any qualitative effect on their conclusions.

This long-standing error could have been prevented by better API design. To model a population split currently in msprime, a user must specify a MassMigration event that moves lineages from one population to another and then must also remember to turn off migration between those populations at the same time. The release of msprime version 1.0 will introduce a PopulationSplit event, which more intuitively links the merger of lineages with appropriate changes in migration rates at the time of the split.

3 Risk prediction in human populations

In another publication using this model (Martin et al., 2017), a separate error was introduced: the model itself was defined as suggested in the documentation (using updated parameters from Gravel et al. (2011)) and inspected using the msprime debugging tools. After these initial checks were made, however, the simulation was performed without passing the parameter of demographic events, so that the three populations remained separated with low levels of migration and never merged as expected (Fig 2A), leading to a vast overestimate of the divergence across human populations. While the correct model predicts a mean F_{ST} of $0.05 - 0.10$ across the three population, the simulated model generated F_{ST} ranging between $0.3 - 0.6$, depending on the populations considered. Overall diversity was also strongly affected: expected heterozygosity was more than doubled in African populations but just $30 - 40\%$ the expected levels in Eurasian populations when compared to diversity under the correct model.

This simulation was performed to assess the transferability of polygenic risk scores across human populations, in particular to explore how human demographic history and population structure affect our ability to predict genetic risk in populations of different ancestry than the originally studied population. The resulting publication has been influential in the discussion of health inequalities and genomics, with over 350 citations since 2017. But while the first implementation error had a small effect on resulting diversity statistics, this error more strongly impacts simulated diversity and interpretations based on that data. The excess divergence under the incorrect model here exaggerated the role that demography plays in limiting the transferability of genetic risk scores across populations.

Difficulties in transferability remain in the corrected model (Fig. 2), although risk prediction in each population is significantly improved when using the correct demographic model (compare to Fig. 5 in Martin et al. (2017)). The corrected simulations indicate that the accuracy of genetic risk scores is still substantially reduced in understudied populations (Fig. 2E-G), supporting the main conclusion of Martin et al. (2017). However, the reduction is much less pronounced than reported. In particular, we do not observe large differences in mean risk prediction across populations (Fig. 2C,D), as presented in Martin et al. (2017). At least for the neutral polygenic architectures considered here, genetic drift alone does not appear to induce large directional biases in mean predicted risk across ethnicities.

4 Conclusions

The implementation of complex demographic models is error prone, and such errors can have a large impact on downstream analysis and interpretation. The discovery and correction of the demographic models discussed here underscore how API design choice can lead to the propagation of mistakes that are difficult to notice. We therefore recommend the following steps to ensure more robust simulations.

First, the design of user interface and API for scientific software matters, and many bugs can be prevented by using more intuitive interfaces. Whereas the original `msprime` required the user to pass three separate parameters to specify a demographic model, `msprime` ver. 1.0 will introduce a Demography class, which wraps these three parameters. Thus, instead of writing

```
dbg = msprime.DemographyDebugger(
  population_configurations=population_configurations,
  migration_matrix=migration_matrix,
  demographic_events=demographic_events)
dbg.print_history()
ts = msprime.simulate(
  population_configurations=population_configurations,
  migration_matrix=migration_matrix,
  demographic_events=demographic_events)
```

with the error-inducing need to re-enter parameter information, we would now write

```
demography = msprime.Demography(
  population_configurations=population_configurations,
  migration_matrix=migration_matrix,
  demographic_events=demographic_events)
dbg = demography.debug()
dbg.print_history()
ts = msprime.simulate(demography=demography)
```

Second, from a user's perspective, steps should be taken to validate the implemented demographic model. This could include verification through code review or an independent implementation of the model, which is the approach taken by `stdpopsim` (Adrion et al., 2019) to compile quality-controlled models and simulation resources for a growing number of commonly studied species. Basic statistical validation should also be performed on any simulated model, and recent progress in computing summary statistics from tree sequence data can make this easier (Ralph et al., 2020). If simulations are so large that statistical validation is burdensome, subsets of the simulated data can be analyzed to ensure general diversity patterns are sound. Such tests likely would have caught the error in Martin et al. (2017).

Finally, openness is essential to the self-correcting nature of science. We only know about these errors because of open code and open-source development processes. By making their entire pipeline available, Martin et al. (2017) not only enabled other research teams to build upon their findings, but they made it possible for such errors to be found and corrected. There must be many, many more mistakes out there, and we need both pre- and post-publication vigilance from users and developers to ensure the soundness of the large body of simulation-based analyses.

5 Methods

We computed the expected allele frequency spectrum using `moments` ver. 1.0.3 (Jouganous et al., 2017), and LD-decay curves using `moments.LD` (Ragsdale and Gravel, 2019). F_{ST} and other diversity statistics were computed from the expected AFS and verified using branch statistics from the tree sequence records of `msprime` simulations (Ralph et al., 2020). Demographic models were plotted using the `demography` package (`https://github.com/apragsdale/demography`, ver. 0.0.3). We used the original pipeline from Martin et al. (2017) available from `https://github.com/armartin/ancestry_pipeline/blob/master/simulate_prs.py`. We updated the pipeline to run with the correct demographic parameters and more recent versions of `msprime` and `tskit` and the updated pipeline is available at `https://github.com/apragsdale/PRS`. Data and python scripts to recreate Figures 1 and 2 can be found at `https://github.com/jeromekelleher/msprime-model-errors`. A full list of the GitHub repositories containing copies of the erroneous model are also given here.

Acknowledgements

- Alicia Martin
- stdpopsim as a whole or anyone in particular from that group?
- others?

References

Jeffrey R Adrion, Christopher B Cole, Noah Dukler, Jared G Galloway, Ariella L Gladstein, Graham Gower, Christopher C Kyriazis, Aaron P Ragsdale, Georgia Tsambos, Franz Baumdicker, et al. A community-maintained standard library of population genetic models. *bioRxiv*, 2019.

Patrick K Albers and Gil McVean. Dating genomic variants and shared ancestry in population-scale sequencing data. *PLoS biology*, 18(1):e3000586, 2020.

Simon Gravel, Brenna M Henn, Ryan N Gutenkunst, Amit R Indap, Gabor T Marth, Andrew G Clark, Fuli Yu, Richard A Gibbs, Carlos D Bustamante, 1000 Genomes Project, et al. Demographic history and rare allele sharing among human populations. *Proceedings of the National Academy of Sciences*, 108(29):11983–11988, 2011.

Ryan N Gutenkunst, Ryan D Hernandez, Scott H Williamson, and Carlos D Bustamante. Inferring the joint demographic history of multiple populations from multidimensional snp frequency data. *PLoS genetics*, 5(10):e1000695, 2009.

Richard R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.

Julien Jouganous, Will Long, Aaron P Ragsdale, and Simon Gravel. Inferring the joint demographic history of multiple populations: beyond the diffusion approximation. *Genetics*, 206(3):1549–1567, 2017.

Jerome Kelleher and Konrad Lohse. Coalescent simulation with msprime. In Julien Y. Dutheil, editor, *Statistical Population Genomics*, pages 191–230. Springer US, New York, NY, 2020.

Jerome Kelleher, Alison M Etheridge, and Gilean McVean. Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS computational biology*, 12(5):e1004842, 2016.

Jerome Kelleher, Yan Wong, Anthony W Wohns, Chaimaa Fadil, Patrick K Albers, and Gil McVean. Inferring whole-genome histories in large population datasets. *Nature genetics*, 51(9):1330–1338, 2019.

Alicia R Martin, Christopher R Gignoux, Raymond K Walters, Genevieve L Wojcik, Benjamin M Neale, Simon Gravel, Mark J Daly, Carlos D Bustamante, and Eimear E Kenny. Human demographic history impacts genetic risk prediction across diverse populations. *The American Journal of Human Genetics*, 100(4):635–649, 2017.

Dominic Nelson, Jerome Kelleher, Aaron P Ragsdale, Claudia Moreau, Gil McVean, and Simon Gravel. Accounting for long-range correlations in genome-wide simulations of large cohorts. *PLoS genetics*, 16 (5):e1008619, 2020.

Peter Pfaffelhuber, Franziska Grundner-Culemann, Veronika Lipphardt, and Franz Baumdicker. How to choose sets of ancestry informative markers: A supervised feature selection approach. *Forensic Science International: Genetics*, 46:102259, 2020.

Aaron P Ragsdale and Simon Gravel. Models of archaic admixture and recent history from two-locus statistics. *PLoS genetics*, 15(6):e1008204, 2019.

Peter Ralph, Kevin Thornton, and Jerome Kelleher. Efficiently summarizing relationships in large samples: a general duality between statistics of genealogies and genomes. *Genetics*, 2020.

Dominic MH Tong and Ryan D Hernandez. Population genetic simulation study of power in association testing across genetic architectures and study designs. *Genetic epidemiology*, 44(1):90–103, 2020.

Ying Zhou, Xiaowen Tian, Brian L Browning, and Sharon R Browning. POPdemog: visualizing population demographic history from simulation scripts. *Bioinformatics*, 34(16):2854–2855, 2018.

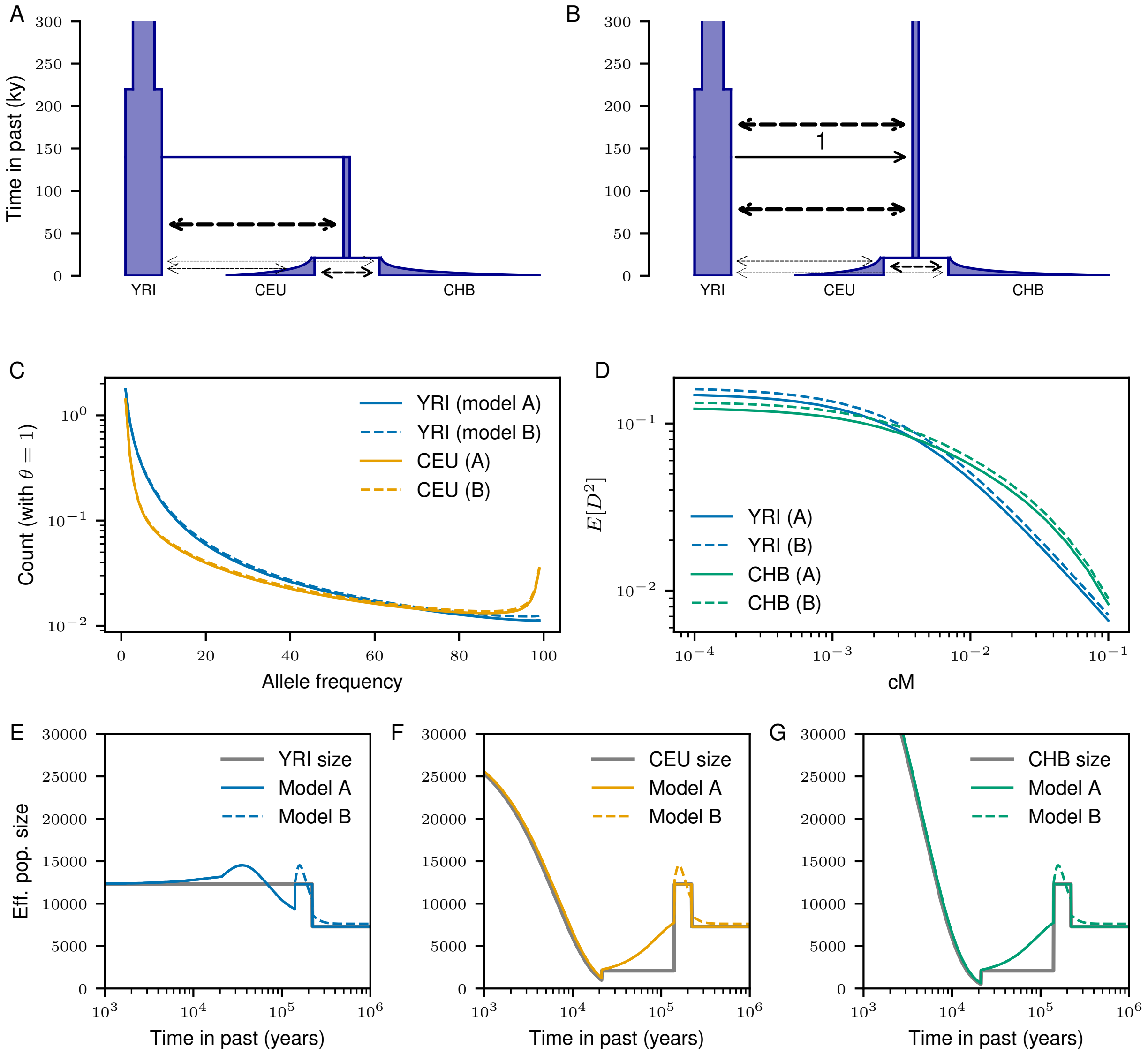

Figure 1: **Expected diversity statistics under the Gutenkunst et al. (2009) model**. **(A)** The correctly implemented model. **(B)** The incorrectly implemented model from the `msprime` tutorial, with migration continuing into the past beyond the mass migration event with proportion 1 from the ancestral population to the bottleneck population. **(C)** Marginal allele frequency spectra under the two models. Heterozygosity in the incorrect model is inflated by 3.5%, though the general shape of the distributions are qualitatively similar. **(D)** Similarly, the increased heterozygosity leads to excess D^2, though the LD-decay is qualitatively similar between models. **(E-G)** True size history for each population plotted against the expected size history from the expected inverse coalescence rates.

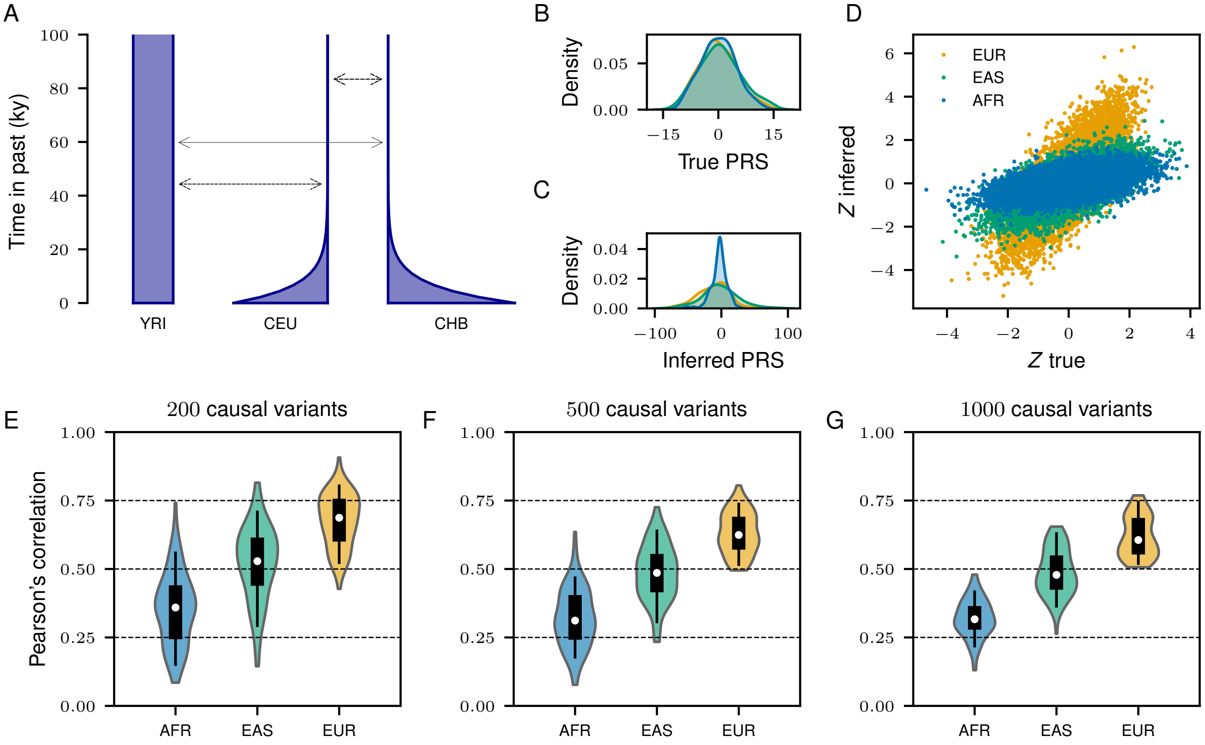

Figure 2: **The transferability of PRS under neutrality**. **(A)** In Martin et al. (2017), the simulated demographic model did not apply demographic events in the past, so continental populations were simulated as isolated with low levels of migration for all time. The intended model is shown in Figure 1A. **(B-G)** We repeated the simulation experiment in Martin et al. (2017) using the correct demographic model. While risk prediction in the African and East Asian population is reduced compared to the studied European population, the reduction in prediction accuracy is not as hopeless as originally reported. Notably, under a neutral polygenic architecture, we do not expect significant bias in inferred PRS in any of the populations (C, D). Correlations were computed over 100 simulation replicates. For direct comparison to the original study, see Figure 5 in Martin et al. (2017).